UCLouvain

linfo2355

2025

Multicore programming

La version que vous consultez n'est pas définitive. Cette fiche d'activité peut encore faire l'objet de modifications. La version finale sera disponible le 1er juin.

5.00 crédits	30.0 h + 15.0 h	Q2

Langue d'enseignement	Anglais > Facilités pour suivre le cours en français
Lieu du cours	Louvain-la-Neuve
Thèmes abordés	Ce cours traite d'un sujet avancé spécifique ou d'une sélection des sujets d'intérêt actuellement au niveau de la recherche dans le domaine du génie logiciel. Le sujet (s) peut (peuvent) varier d'année en année, et sera choisi parmi une variété de domaines d'ingénierie logicielle tels que l'informatique de données à forte intensité, l'analyse de logiciels, le développement et l'analyse de grands systèmes évolutifs de logiciels, les techniques liées aux grandes quantités de données, la fouille du code des logiciels , les systèmes de recommandation de logiciels, la visualisation de logiciels, les nouvelles technologies de programmation, les exigences de logiciels et l'analyse logicielle, l'ingénierie logicielle basée sur des modèles, la gestion de la configuration de logiciel, les processus d'ingénierie logicielle, les outils et méthodes de génie logiciel, les tests et la qualité des logiciels, etc.
Acquis d'apprentissage	**A la fin de cette unité d'enseignement, l'étudiant est capable de :** 1 Eu égard au référentiel AA du programme « Master ingénieur civil en informatique », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants : • INFO1.1 • INFO3.1 • INFO6.3 Eu égard au référentiel AA du programme « Master [120] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants : • SINF1.M3 • SINF3.1 • SINF6.3 Les élèves acquièrent des connaissances théoriques de pointe et des compétences techniques sur les sujets abordés dans le cours.

Modes d'évaluation des acquis des étudiants	Modes d'évaluation :
	• Projets et contrôle continu (40% de la note finale)
	• Examen sur table comportant deux parties (60% de la note finale)
	  • Une première de questions de cours (questions ouvertes et QCM)
	  • Une seconde partie portant sur un ou plusieurs articles de recherche du domaine préalablement fournis aux étudiant·es.
	  • La proportion entre les deux parties est variable mais aucune ne dépassera 70% de la note. Les QCM n'entraînent pas de points négatifs pour une question individuelle mais la section QCM dans son ensemble peut demander un seuil minimal de réponses correctes avant de valoir des points.
	
	Pour la seconde session (Aout), le projet et l'évaluation continu peuvent être refait mais uniquement de façon individuelle.
	L'évaluation des travaux soumis (projets, travaux pratiques, travail individuel, et de manière générale toute activité donnant lieu à évaluation) est basé sur la matériel (code, rapports, et tout autre matériel) soumis par l'étudiant·e, accompagné sur demande du profeseur par un examen oral complémentaire.
	Sauf indication explicite tout travail demandé doit rester confidentiel et personnel. Le partage d'information et/ou de code avec d'autres étudiants (hors binôme le cas échéant) constitue un cas de plagiat.

Méthodes d'enseignement	• Cours magistraux • Lectures et / ou production de courtes vidéos • Séances pratiques • Mini-projet

Contenu	Ce cours aborde la problématique de la programmation passant à l'échelle pour les architecture multi-processeurs et multi-cœurs modernes. Dans une courte première partie, le cours aborde les éléments de théorie et les modèles de cohérence, et permet la compréhension des enjeux liés à la programmation concurrente en environnement en mémoire partagée. L'emphase est mise sur les aspects liés à la performance et au passage à l'échelle (utilisation de nombreux cœurs). Le reste du cours passe en revue un ensemble de techniques algorithmiques pour la construction de structures de données concurrentes en mémoire partagée passant à l'échelle, et étudie leurs implications en termes de performance sur les architectures modernes, en prenant en compte la hiérarchie mémoire, les protocoles de cohérence, ou les accès mémoire non uniformes (NUMA). Le cours est accompagné d'une série de travaux pratiques sur une machine multi-cœurs, au cours desquels les étudiant·e·s évaluent la performance et le coût de divers algorithmes et structures de données concurrents.
Bibliographie	The Art of Multiprocessor Programming, Maurice Herlihy and Nir Shavit, Morgan Kaufmann. ISBN 978-0-12-370591-4. UCL library reference 10.620.426
Autres infos	Tout le matériel de cours pertinent et les diapositives, ainsi que des informations pratiques relatives au cours seront accessibles sur Moodle, qui sera également le principal moyen de communication entre l'enseignant (s) et les étudiants.
Faculté ou entité en charge:	INFO

Programmes / formations proposant cette unité d'enseignement (UE)

Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Master [120] : ingénieur civil en informatique	INFO2M	5		
Master [120] en sciences informatiques	SINF2M	5		
Master [120] : ingénieur civil en science des données	DATE2M	5		
Master [120] en science des données, orientation technologies de l'information	DATI2M	5		